

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2001-169246

(43)Date of publication of application : 22.06.2001

(51)Int.Cl.

H04N 5/92
G11B 20/12
G11B 27/00
G11B 27/031
H04N 5/91

(21)Application number : 11-352581

(71)Applicant : SHARP CORP

(22)Date of filing : 13.12.1999

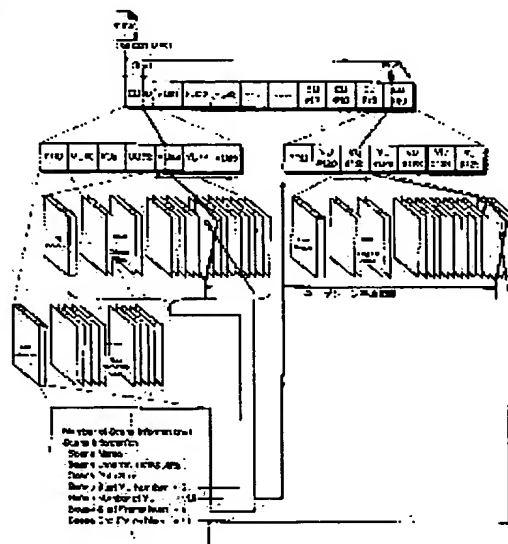
(72)Inventor : IWANO HIROTOSHI

(54) DATA RECORDING METHOD AND DATA REPRODUCTION METHOD

(57)Abstract:

PROBLEM TO BE SOLVED: To solve a problem involved in a conventional data recording method and data reproduction method that complicated processing procedures has been required because it is required to transfer two files for stream data being part of an original scene and its management information if the stream data corresponding to a user scene defined by a user is copied or transferred to a network due to the absence of concept of nondestructive edit caused by e.g. one user scene consisting of one independent file despite of the necessity of reproduction management information for each user scene or due to the reproduction management information intensively recorded on a disk as a file separately from a file of stream data even when the nondestructive edit is preconditioned.

SOLUTION: Information relating to a user scene is recorded in a header position of each data unit in stream data so as to manage reproduction information together with the stream data and even a data transfer destination can control a reproduction start and end from an optional frame position.



(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2001-169246

(P2001-169246A)

(43) 公開日 平成13年6月22日 (2001.6.22)

(51) Int.Cl. ⁷	識別記号	F I	テーマコード* (参考)
H 0 4 N 5/92		G 1 1 B 20/12	5 C 0 5 3
G 1 1 B 20/12			1 0 3 5 D 0 4 4
	1 0 3	27/00	A 5 D 1 1 0
27/00		H 0 4 N 5/92	H
27/031		5/91	Z

審査請求 未請求 請求項の数9 O L (全 17 頁) 最終頁に続く

(21) 出願番号 特願平11-352581

(22) 出願日 平成11年12月13日 (1999. 12. 13)

(71) 出願人 000005049

シャープ株式会社

大阪府大阪市阿倍野区長池町22番22号

(72) 発明者 岩野 裕利

大阪府大阪市阿倍野区長池町22番22号 シ

ャープ株式会社内

(74) 代理人 100103296

弁理士 小池 隆彌

F ターム (参考) 5C053 FA14 FA23 FA27 GB37 HA29

JA03 LA01 LA11 LA14

5D044 AB05 AB07 BC06 CD04 DE03

DE17 DE48 DE52 HL14

5D110 AA17 CA05 CA06 CA07 CD01

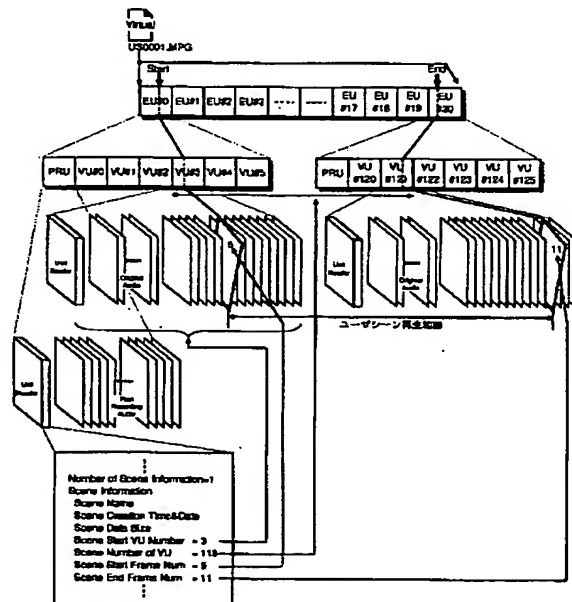
DA11 DB02 DE01

(54) 【発明の名称】 データ記録方法及びデータ再生方法

(57) 【要約】

【課題】 ユーザシーン毎に再生管理情報が必要になるが、従来の方法では、例えば1つのユーザシーンが1つの独立したファイルで構成され非破壊編集の概念がなかったり、非破壊編集を前提とした場合であっても必要となる再生管理情報が集中的にストリームデータとは別のファイルとしてディスクに記録されているため、仮にユーザによって定義されたユーザシーンに対応するストリームデータをコピーやネットワーク転送する場合は、オリジナルシーンの一部であるストリームデータとその管理情報の2つのファイルを転送する必要があり、処理手順が煩雑になってしまうという問題点がある。

【解決手段】 ストリームデータ中の各データユニットのヘッダ位置にユーザシーンに関する情報を記録することにより、ストリームデータとともに、再生情報を管理することができ、データ転送先においても、任意のフレーム位置からの再生開始、終了を制御することを可能とする。



【特許請求の範囲】

【請求項 1】 記録媒体上に、1つあるいは複数のGOPデータ及び対応する音声データ、及び補助データで構成されるユニットの集合であるファイルとして映像データを記録し、該ファイル単位で映像データの読出し及び書き込みを行う記録再生装置におけるデータ記録方法であって、
当該ファイルの先頭のユニットの補助データに、当該ファイルの再生情報を記録することを特徴とするデータ記録方法。

【請求項 2】 記録媒体上に、1つあるいは複数のGOPデータ及び対応する音声データ、及び補助データで構成されるユニットの集合であるファイルとして映像データを記録し、該ファイル単位で映像データの読出し及び書き込みを行う記録再生装置におけるデータ記録方法であって、

記録されたファイル中の全部あるいは一部のユニットを仮想ファイルとして管理し、

当該仮想ファイルの先頭のユニットの補助データに、当該ファイルの再生情報を記録することを特徴とするデータ記録方法。

【請求項 3】 前記再生情報は、再生を開始するフレーム及び再生を終了するフレームを示す情報を含むことを特徴とする前記請求項 1 または 2 に記載のデータ記録方法。

【請求項 4】 前記再生情報は、当該ファイル或いは仮想ファイルのファイルサイズを含むことを特徴とする前記請求項 1 乃至 3 のいずれかに記載のデータ記録方法。

【請求項 5】 前記ユニットを 1 GOP 単位とし、前記再生情報を MPEG フォーマットにおける GOP 層のユーザデータ領域に記録することを特徴とする前記請求項 1 乃至 4 のいずれかに記載のデータ記録方法。

【請求項 6】 1つあるいは複数のGOPデータ及び対応する音声データ、及び補助データで構成されるユニットの集合であるファイルとして映像データが記録された記録媒体において、該ファイル単位で映像データの読出し及び書き込みを行う記録再生装置におけるデータ再生方法であって、

読み出されたファイルの先頭のユニットの補助データに記録されている再生情報を読み出し、当該再生情報に基づいて、読み出されたファイルの再生制御を行うことを特徴とするデータ再生方法。

【請求項 7】 前記再生情報は、再生を開始するフレーム及び再生を終了するフレームを示す情報を含むことを特徴とする前記請求項 6 に記載のデータ再生方法。

【請求項 8】 前記再生情報は、管理領域に記録される当該ファイルのファイルサイズを含み、再生するファイルのファイルサイズと、一致する再生情報を選択し、選択された再生情報に基づいて、再生制御を行うことを特徴とする前記請求項 6 または請求項 7 に

記載のデータ再生方法。

【請求項 9】 前記ユニットは 1 GOP 単位であり、前記再生位置管理情報は、MPEG フォーマットにおける GOP 層のユーザデータ領域に記録されていることを特徴とする前記請求項 6 乃至 8 のいずれかに記載のデータ記録方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、動画ファイルの記録再生する方法に関するものであり、特に非破壊編集を行う場合における再生管理情報の記録手法に関する。

【0002】

【従来の技術】近年のマルチメディアの普及に伴い、映像、音楽、静止画などの様々なマルチメディアデータを、記録媒体へ記録する需要が高まってきている。記録媒体の中でも、従来はビデオテープやオーディオテープなどのテープメディアが主流であったが、近年はハードディスク、光磁気ディスクなどのディスクメディアに記録することが多くなってきている。ディスクメディアを利用したものとして、音楽用の場合は MD、映像用の場合は DVD Video などが一般的に知られており、ランダムアクセス性を特徴として普及している。

【0003】ここで、MPEG でエンコードした映像データと音声データを多重したストリームデータをディスクに記録する場合について説明する。例えば、ディスクに記録したストリームデータを記録開始から停止あるいは一時停止と言った管理単位でファイルとして管理する事にする。ディスクのランダムアクセス性を利用すると、1つのファイルに対応するストリームデータであってもディスク上で連続的に記録されている必要はない。つまり1つのファイルで管理する一連のデータがディスク上で分断して記録されていても、ディスクからそれらの分断を順番に読み出して行くことによって、各分断点において次にデータを読み出すディスク上の位置までディスク装置のヘッドを移動させるシークやトラックジャンプが発生するが、テープメディアと比較して無視できるデータ読み出し中断時間なので、あたかも連続的に対応するデータをディスクから読み出しているのと同様の効果が得られる。

【0004】以下、本実施例においてユーザによって記録された記録開始から終了あるいは一時停止の一連のストリームデータをオリジナルシーンと呼び、ユーザが編集操作によってオリジナルシーンの任意の箇所を選択したストリームデータをユーザシーンと呼ぶ事とする。また、全てのオリジナルシーンを組み合わせた管理単位をオリジナルプログラムと呼び、任意の数のユーザシーンを任意の順番で組み合わせた管理単位をユーザプログラムと呼ぶ事とする。

【0005】ディスクのランダムアクセス性は、編集においても効果を発揮する。ディスクメディアに記録され

10

20

30

40

50

たストリームデータを編集する場合について説明する。ビデオカメラなどを考えた場合、撮影したオリジナルシーンのストリームデータは必ずしも全て見たいものであるとは限らない。例えば、オリジナルシーンの最初に不要な部分が含まれている事も考えられる。編集操作を行なうことによって、必要な任意の箇所をオリジナルシーンから選択してユーザシーンとして定義することが可能である。ランダムアクセス性を利用すると、素材データであるオリジナルシーンのストリームデータをコピーしたり手を加えることなくディスク上のストリームデータを共有する形で非破壊編集を行ない、ユーザシーンに対応するストリームデータのディスク上の位置情報を元に再生が可能となる。このように、オリジナルシーンに対応するディスク上のストリームデータを参照する形で定義されるユーザシーンを、仮想ファイルとして管理するものとする。

【0006】図24に示すように3つのオリジナルシーンに対応するストリームデータがディスク上に記録されており、それぞれのオリジナルシーンにおいて任意の箇所を選択し3つのユーザシーンが定義されている例について説明する。この例では、ディスク上のオリジナルシーンのストリームデータは論理ファイルシステムによって、ファイル名OS0001.MPG、OS0002.MPG、OS0003.MPGとディスク上の記録位置が関連付けられ管理されている。

【0007】一方、ユーザシーンは論理ファイルシステムによって仮想ファイルとしてUS0001.MPG、US0002.MPG、US0003.MPGという名前で管理されている。仮想ファイルの論理ファイルシステムの管理情報は参照しているオリジナルシーンのファイルを特定するための情報とストリームデータのディスク上での選択箇所を示す開始点と長さの集合である。仮想ファイルのポインタ情報によってオリジナルシーンの任意の箇所を抜き出し、仮想的なファイルとして扱うことが可能となる。

【0008】ファイルや仮想ファイルの管理情報は論理ファイルシステム階層のものであるため、デバイスドライバを介してアクセスすることによって、ディスク上にオリジナルシーンのデータが記録されているだけに関わらず、ユーザシステムから見ると仮想ファイルで管理されるユーザシーンに対応するデータが仮想的に別途ディスクに記録されているものとして扱われる。つまり、仮想ファイルUS00001.MPGをディスクから読み出す命令をデバイスドライバに渡すことによって、ユーザシーンを構成するオリジナルシーン中の選択箇所のデータが自動的に読み出されることになる。

【0009】このようにオリジナルシーンをファイルで、ユーザシーンを仮想ファイルで管理することによって、例えば編集したユーザシーンやユーザプログラムを他の人にあげる用途などで、IEEE1394などのネットワークで接続されたコンピュータなどに転送の基本コマンドであるファイルのGetやPutのみで転送することが可能と

なる。これは、転送を行ないたいオリジナルシーンを特定するファイル名やユーザシーンを特定するための仮想ファイル名を指定して転送を行なうことによって達成されるものである。

【0010】しかしながら、ファイルや仮想ファイルの枠組みでMPEGストリームを管理する場合に問題になることがある。MPEG技術では、データ量を圧縮するにあたって、その映像フレームのデータだけで独立して符号化するフレーム内符号化画像(Iピクチャ)、前方向のフレームの情報を元に符号化するフレーム間順方向予測符号化画像(Pピクチャ)、前方向と後方向のフレームを元に符号化する双方向予測符号化画像(Bピクチャ)という3種類の画像圧縮手法を使って、効率的にデータ量を削減している。

【0011】つまり、PピクチャやBピクチャを再生したい場合は、レファレンスとなったIピクチャやPピクチャのデータがないと、デコードできないことを意味する。そこで、このような問題を解決するために、MPEGにおいては、何枚かのフレームを集めたGOP(Group of Pictures)という構造が用意されている。このGOP構造は、GOPの中には少なくとも1枚のIピクチャが含まなければならないというものである。従って、GOP構造単位でアクセスを行なえば、そのGOPの中に含まれている各Pピクチャ及びBピクチャのレファレンスとなるIピクチャが含まれているので、目的のフレームをデコードすることが保証される。

【0012】このように、MPEGストリームデータを対象にランダムアクセスを行なう場合は、GOP構造に相当する単位で行なう必要がある。つまり、ファイルや仮想ファイルで管理するストリームデータの最小管理単位はGOPに相当する管理単位であり、ファイルや仮想ファイルを構成するストリームデータはGOP構造に相当するストリームデータの管理単位を整数個集めた構造となることになる。

【0013】仮に1GOPが15フレームの映像データで構成されているとすると、ファイルや仮想ファイルによって管理されるストリームデータは15フレーム単位のストリームデータということになり、ファイルや仮想ファイルを指定して再生を行うと、このGOP単位の再生になる。

【0014】しかしながら、一般的に、ユーザが再生したいユーザシーンがGOPの先頭から開始されることはまれであり、不要なフレームも再生されてしまうという問題がある。一般的に図25に示すようにユーザシーン情報をストリームデータ自身の先頭にヘッダとして記録する方法がある。この場合は、ユーザによる編集結果であるユーザシーンは仮想ファイルでは管理せず、ユーザシーン毎に独立した実ファイルで管理する事になる。つまり、編集結果であるユーザシーン毎にディスク上に独立したストリームデータが記録されることになり、実ファイルの一部を仮想ファイルとするのに比べ、ディスク領

域を多く消費することになる。これは、本発明が前提としている非破壊編集ではない。

【0015】また、図26に示すように非破壊編集を前提として、ディスクに記録されたストリームデータを1つの実ファイルで管理し、そのストリームデータ内の詳細は別の実ファイルとしてディスクに記録される管理情報で管理する方法がある。図の例では、再生情報が格納されている実ファイルと、ストリームデータが格納されている実ファイルの2つのファイルと、論理ファイルシステムの管理情報として、ファイル名とディスク上のデータの記録位置を関連付けるファイル管理情報がディスク上に記録されている。ここでの再生情報とは、各ユーザシーンに対応するストリームデータがファイルで管理されているデータのどの箇所のものであるかを特定するための情報と、前述したようにディスクからの読み出しがGOPに相当する単位になるため、再生を開始および停止するフレームを特定するための情報である。

【0016】

【発明が解決しようとする課題】オリジナルシーンの任意の箇所を参照する形で行われる非破壊編集によって定義されるユーザシーンについて、対応するディスク上のストリームデータを他の機器へコピーしたりネットワーク転送などを行う事を考えると、従来技術で述べた方法などによって実現できた。

【0017】しかしMPEG技術によって記録されたディスク上のストリームデータに関して任意の箇所を選択し抜き出すユーザシーンを他の機器などにコピーや転送する場合、対応するストリームデータが前述したようにGOPに相当する管理単位毎に行われる必要がある。よって転送先において、フレーム単位の再生範囲を示す管理情報がなければ、実際にコピーや転送されたGOP単位での再生しかできなくなる。また、一般的に転送先において単純な映像再生を行うだけであれば、単純にストリームデータだけあれば十分であるが、それらに付随するタイトルやデータに関する属性情報なども一緒に転送する事によって、単純な再生だけではなく様々な処理を行う事が可能となる。

【0018】このように、ユーザシーン毎に再生管理情報が必要になるが、従来の方法では、例えば1つのユーザシーンが1つの独立したファイルで構成され非破壊編集の概念がなかったり(図25)、非破壊編集を前提とした場合であっても必要となる再生管理情報が集中的にストリームデータとは別のファイルとしてディスクに記録されている(図26)。よって、仮にユーザによって定義されたユーザシーンに対応するストリームデータをコピーやネットワーク転送する場合は、オリジナルシーンの一部であるストリームデータとその管理情報の2つのファイルを転送する必要が出てくる。転送先においてストリームデータと管理情報が2つのファイルに別れているのは、例えば転送先がPCでありユーザによって誤っ

て管理情報のファイルが消えてしまうと再生情報が消失してしまう事も考えられ問題がある。

【0019】そこで、コピーやネットワーク転送を行う場合には極力関連するデータを1つのファイルとして行うのが望ましいが、これを行うためにはコピーや転送を行う時に、(全てのユーザシーンに関する再生情報が)1つのファイルとして集中管理されている管理情報から目的のユーザシーンに関する再生情報を抜き出し、コピーや転送しようとするストリームデータのヘッダ情報として付加する処理を行う必要がある。よって、ユーザシーンなどをコピーやネットワーク転送する際には、前処理を行う必要が生じ処理手順が煩雑になってしまうという問題点がある。

【0020】そこで、本願発明においては、ストリームデータ中の各データユニットのヘッダ位置にユーザシーンに関する情報を記録することにより、ストリームデータとともに、再生情報を管理することができ、データ転送先においても、任意のフレーム位置からの再生開始、終了を制御することを可能としたり、単純な再生だけではなく様々な処理を行う事が可能とするものである。

【0021】

【課題を解決するための手段】本願の第1の発明によれば、記録媒体上に、1つあるいは複数のGOPデータ及び対応する音声データ、及び補助データ(Unit Header)で構成されるユニット(VU/PRU)の集合であるファイル(EUS)として映像データを記録し、該ファイル単位で映像データの読み出し及び書き込みを行う記録再生装置におけるデータ記録方法であって、当該ファイルの先頭のユニット(VU/PRU)の補助データ(Unit Header)に、当該ファイルの(再生開始位置及び終了位置を示す)再生情報を記録することにより、上記課題を解決する。

【0022】また、本願の第2の発明によれば、記録媒体上に、1つあるいは複数のGOPデータ及び対応する音声データ、及び補助データ(Unit Header)で構成されるユニット(VU/PRU)の集合であるファイル(EUS)として映像データを記録し、該ファイル単位で映像データの読み出し及び書き込みを行う記録再生装置におけるデータ記録方法であって、記録されたファイル中の全部あるいは一部のユニットを仮想ファイルとして(例えばVirtual File Descriptorなどで)管理し、当該仮想ファイルの先頭のユニットの補助データ(Unit Header)に、当該ファイルの(再生開始位置及び終了位置を示す)再生情報を記録することにより上記課題を解決する。

【0023】本願の第3の発明によれば、1つあるいは複数のGOPデータ及び対応する音声データ、及び補助データで構成されるユニット(VU/PRU)の集合であるファイル(EUS)として映像データが記録された記録媒体において、該ファイル単位で映像データの読み出し及び書き込みを行う記録再生装置におけるデータ再生方法であって、読み出されたファイルの先頭のユニット(VU/PRU)の補助

10

20

30

40

50

データ(Unit Header)に記録されている(再生開始位置及び終了位置を示す)再生情報を読み出し、当該再生情報に基づいて、読み出されたファイルの再生制御を行うことにより上記課題を解決する。

【0024】

【発明の実施の形態】以下、本発明のファイル管理方法に関する実施形態について、図面を用いて詳細に説明する。本実施形態において、記録装置として携帯型のディスクを用いたビデオカメラを、ディスクに記録する映像データはMPEGを想定する。また、ディスク装置に関しては、据え置き型のビデオデッキや、記録媒体はハードディスクや半導体メモリであっても本実施形態をそのまま適用できるものである。

【0025】以下、本実施形態の説明において、ディスクに記録されたオリジナルシーンに対応するディスク上のオリジナルデータを管理するファイルを実ファイルと呼び、オリジナルデータを参照する形で構成されるユーザシーンを管理するファイルを仮想ファイルと呼び、区別して説明するものとする。

【0026】ユーザが撮影した記録開始から停止あるいは一時停止までのオリジナルシーンを論理ファイルシステムにおいて実ファイルとして扱い、その素材データであるオリジナルシーンのデータをコピーしたり手を加えることなくディスク上のストリームデータを共有する形(いわゆる非破壊編集)で、オリジナルシーンの任意の箇所を選択することによって定義されるユーザシーンを論理ファイルシステムの仮想ファイルで管理を行うものとする。既にディスク上で実ファイルとして管理されているデータの任意の箇所を共有するため仮想ファイルと呼ぶものである。

【0027】図1に、本発明のシステム構成図の一例を示す。記録時と再生時の処理の流れに併せて説明を行なう。まず記録時の説明を行なう。ユーザから記録要求を受けとった制御部1は、各処理部に対して制御信号を出しシステム全体を制御する。カメラ部2からの映像と音声入力はまずMPEGエンコーダ3においてそれぞれエンコードされる。それぞれのエンコードされたデータはMPEGシステム部4において映像と音声が多重され、例えばMPEGのPESストリームなどのストリーム構成に整形されることになる。このストリームデータは一時的にバッファメモリ5に格納される。そして、ECC信号処理部6がバッファメモリ5に格納されたストリームデータに対してECC(Error Correction Code)符号を付加したりする信号処理を施す。ECCなどの信号処理が行なわれたストリームデータはディスク9に記録するために、変復調/セクタコーディック部7で、変調及びセクタ構造に併せてフォーマットを行い、サーボ制御部8がディスク9を制御して記録されることになる。

【0028】続いて、再生時の処理を説明する。ユーザから再生要求を受けとった制御部1は、各処理部に対

して制御信号を出しシステム全体を制御する。ディスク9から目的のストリームデータがサーボ制御部8の制御下で読み出されたら、変復調/セクタコーディック部7で復調され、復調されたストリームデータはバッファメモリ5に格納される。バッファメモリ5に格納されたストリームデータはECC信号処理部6によって、記録時に付加されたECC符号などにより誤り訂正が行なわれ、余分な符号などが取り除かれる。実際に再生する段階に来た時に、MPEGシステム部4において多重化されていた映像と音声データを分離し、このデータをMPEGデコーダ部3においてデコードし、図示しない実際のモニタ画面等の出力装置に表示されることになる。

【0029】次に、本実施形態で扱うMPEGストリームの構成の一例について説明を行なう。図2のストリーム構成において、EUS(Editable Unit Sequence)は、複数のEU(Editable Unit)によって構成され、REC Start(記録開始)からREC Stop(記録停止)或いはREC Pause(記録一時停止)に対応する単位である。図24に示した、オリジナルシーンに対応するストリームデータがEUSに対応する。

【0030】尚、EUは破壊編集における最小単位である。破壊編集とは、ディスク上での移動や削除を伴う編集のことを意味し、破壊編集の最小単位とは、ディスク上での移動や削除がEU単位でしか行うことできないことを意味する。EUは1つ以上のVU(Video Unit)及び0あるいは1つのPRU(Post Recording Unit)によって構成され、1つのEUはディスク上では必ず連続的に記録されなければならない。尚、PRUが無いストリーム構成も定義することが可能であり、その様子を図3に示す。この例では1つのEUは1つのVUによって構成されている様子である。なおPost Recordingとはアフレコのことを意味する。また、PRUはEU内のビデオデータと同期して再生するPost Recording用のデータ領域であるので、最低でもEUのビデオデータの提示時間に相当するだけのデータが記録できる領域がなければならない。また、VUはUnit Headerと1GOP以上の映像データ及び対応する音声データとをまとめた単位である。

【0031】前記EUSは2048byteの固定長のブロックに分割される。1つのブロックは1つの論理ブロックに格納され、1つのブロックは原則として1個のバケットで構成される。ここでのバケットは、ISO/IEC13818-1で規定されるPES Packetに準拠し、ディスクにはこのバケットを記録していくことになる。

【0032】図4にPRUが存在するEUSとブロックとの関係、図5にPRUが存在しないEUSとブロックとの関係を示す。図中において、PRUはUH-BLK(Unit Header Block)、A-BLK(Audio Block)、P-BLK(Padding Block)で構成される。UH-BLKは、PRUに関するヘッダ情報を格納したバケット、A-BLKは、ISO/IEC13818-3で規定されるオーディオバケット、P-BLKは、ISO/IEC13818-1で規

10

20

30

40

50

定されるパディングバケットがそれぞれ格納される。

【0033】また、VUはUH-BLK (Unit Header Block)、A-BLK (Audio Block)、V-BLK (Video Block) によって構成される。UH-BLKは、VUに関するヘッダ情報を格納したバケット、A-BLKは、ISO/IEC13818-3で規定されるオーディオバケット、V-BLKは、ISO/IEC13818-2で規定されるビデオデータを格納したバケットがそれぞれ格納される。UH-BLKは、PRUあるいはVUに関するヘッダ情報を格納したバケットである。

【0034】Unit Headerはいわゆるユニットに関する補助データ領域である。このUnit Headerバケットの構成を図6に示す。図6の表中の、BPはByte Positionを意味し、先頭から見た対応する管理項目の開始位置を示す情報で、Lengthはその管理項目の大きさをByteで表し、Field Nameは管理項目名、Contentsは、管理項目がどのような形式で記録されなければならないかということを示す。Contentsで用いられているデータ型のうち、Uint8は符号無し8bit整数、Uint16は符号無し16bit整数、Uint32は符号無し32bit整数、Stringは文字列を格納するためのデータ型、Timestampは日時情報を格納する型である。

【0035】Unit Headerバケットは、バケットが始まることを示すpacket-start-code-prefix、ストリームのIDを示すstream-id、このPES Packetの長さを示すPES-packet-length、このPacketが含まれるUnitの状態を示すUnit Property、このPacketの含まれるUnitの長さを示すLength of Unit、このpacketが含まれるunitの先頭からビデオデータの含まれる最初のblockまでの相対論理ブロック数を示すStart RLBN of Video Data、このpacketに含まれるユニット中の全てのI PictureおよびP Pictureの数を示すNumber of IP Pictures、packetに含まれるunit内のI PictureおよびP Pictureの含まれるディスク上の最終アドレスを示すEnd RLBN of IPPictures、このpacketの含まれるEU中に含まれるVUの数を示すNumber of VU、このpacketの含まれるEUに含まれるそれぞれのVUに対応するPRU中のデータの、PRUの先頭からの相対論理ブロック数を示すStart RLBN of Data for VU、管理するシーン数を表すNumber of Scenes、シーンに関する管理情報を管理するSceneInformationで構成される。なお、Scene Informationの詳細については後述する。

【0036】ここで、管理できるScene Informationに数の上限を設ける。例えば、20シーンを上限とした場合、Unit Headerにはあらかじめ20シーン分のScene Informationが記録できるだけの領域を確保するものとする。これはScene Informationがストリームデータをディスクに記録し、ユーザシーンを作成した際にScene Information部分のみを書き換えるために、あらかじめそのデータ領域を用意しておくためのものである。一番初めにストリームデータをディスクに記録する際は、例えばNumber of Scenesに0を、Scene Informationを20個記

録するための領域には全て16進数でFFを記録しておく。

【0037】なお、Unit Header blockは論理ブロックの単位でアライメントされ、Unit Header Blockの先頭は必ず論理ブロックの先頭に一致するように構成されているので、容易にアクセスすることが可能である。

【0038】上記のMPEGストリームをディスクに記録する際の実ファイルで管理されるオリジナルシーンと仮想ファイルで管理されるユーザシーンは、既に述べたようにMPEGにおけるGOPに相当するストリームデータの集合であるVU単位でなければならない。これは、MPEGストリームデータの途中から再生を開始するフレームが、PビクチャやBビクチャの場合、実際にそのフレームをデコードするためには、レファレンスとなったIビクチャやPビクチャのデータが無いとデコードできないからである。また、前述した本実施例におけるPRUの存在するストリーム構成においては、更にGOP構造に相当するストリームデータの集合である任意の個数のVUとPRU領域で構成されるEU単位で、ファイルや仮想ファイルを扱わなければならない。これは、EU内の全てのVUに対応してPRUの領域が連続的に割り当てられており、EU内の各VUとPRUを分けて管理することが困難であるためである。

【0039】PRUの存在しないストリーム構成の場合は、VU単位でファイルや仮想ファイルを作成することになる。よって、オリジナルシーンやユーザシーンを管理するファイルや仮想ファイルは、PRUが存在するストリームデータを対象にするかによって、ディスク上の整数個のEUあるいはVUを集めた管理単位となる。

【0040】図7にオリジナルシーンである実ファイルと、ユーザシーンである仮想ファイルとの関係を示す。この例では、オリジナルシーンは実ファイルOS0001.MPGであり、該ファイルの管理情報で管理されている。OS0001.MPGは、ディスク上で3つの連続領域に分断されて記録されている。オリジナルシーンをディスクから読み出す際は、ユーザシステムがデバイスドライバにファイルOS0001.MPGの読み出しを指定することによって、論理ファイルシステムの管理情報で管理されるディスク上の分断1A、分断2A、分断3Aの位置情報（開始アドレスと長さ）を元に、デバイスドライバが分断1A、分断2A、分断3Aの順番にディスクからデータを読み出す。

【0041】ここで実ファイルで管理されるディスク上のデータと論理ファイルシステムの管理情報の関係を図9に示す。図の例ではディスク上に、1つの実ファイルとして3つに分断して記録されたデータとそれらを管理するためのファイル管理情報が記録されている。このファイル管理情報によって、ファイル名からディスク上に記録されたデータの記録位置を特定する事が可能となる。ファイル管理情報は、通常ディスクにアクセスするためのデバイスドライバが利用する情報であり、ユーザシステムからは直接見えない情報である。つまり、ユーザシステムはデバイスドライバに対してファイル名を指定し

てデータの読み出し命令を出すわけである。本発明では実ファイルを管理するためのファイル管理情報をFile Descriptorと呼ぶ事とする。

【0042】次に、図7においてオリジナルシーンを編集してEU#1からEU#7までを選択して定義されたユーザシーンについて説明する。ユーザシーンは仮想ファイルUS0001.MPGとして管理されており、論理ファイルシステムの管理情報内の分断1B、分断2B、分断3Bの位置情報（開始アドレスと長さ）によってディスク上のオリジナルデータを参照している。ここで、ディスク上にはEU#0からEU#9に対応する映像データが1つしか記録されていないが、仮想ファイルで参照される任意箇所の位置情報によって、部分的なデータの読み出しがユーザシステムにおいては仮想ファイルを指定することによって可能となる。

【0043】ここで仮想ファイルで管理されるディスク上のデータと論理ファイルシステムの管理情報の関係を図10に示す。図の例ではディスク上に記録されたデータを1つの実ファイルで管理し、その実ファイルが管理するデータの任意の箇所を参照する形でそのディスク上の位置を仮想ファイルで管理するものである。ディスク上には、実ファイルおよび仮想ファイルのファイル管理情報が記録されている。これらのファイル管理情報によって、ファイル名からディスク上に記録されたデータの記録位置を特定する事が可能となる。実ファイルのファイル管理情報では、ディスク上に記録されたデータの位置情報を管理し、仮想ファイルのファイル管理情報では、実ファイルが管理するディスク上に記録されたデータの任意の箇所の位置情報を管理するものである。実ファイルのファイル管理情報と仮想ファイルのファイル管理情報はそれぞれを区別するための情報を保有している。本発明では仮想ファイルを管理するためのファイル管理情報をVirtual File Descriptorと呼ぶ事とする。

【0044】このように、仮想ファイルはディスク上に記録されているオリジナルシーンのデータを参照するものである。また、仮想ファイルは同一のオリジナルシーンを複数のユーザシーンで参照することも可能である。図8に2つのユーザシーンを管理する仮想ファイルの様子を示す。この例では、EU#0からEU#9までがオリジナルシーンであり、EU#1からEU#7までがユーザシーン1（US0001.MPG）、EU#6からEU#9までがユーザシーン2（US0002.MPG）という構成である。つまり、EU#6、EU#7が2つのユーザシーンで参照されていることになる。

【0045】PRU領域の定義されないストリームデータの場合についてのオリジナルシーンである実ファイルと、ユーザシーンである仮想ファイルとの関係を図11、12に示す。上記図7、8との違いは、EUがVUとなるだけであるので、説明は省略する。

【0046】このような形態で管理されるユーザシーンを、IEEE1394などを用いてネットワーク経由でPCやその

他のAV機器に転送することを考える。オリジナルシーンやユーザシーンのコピーやネットワーク転送を行なうには、それぞれ対応するファイルや仮想ファイルを指定してファイルコピーを行なったり、ファイル転送を行なうことになる。例えば図24における、仮想ファイルによって管理されるユーザシーンUS0001.MPGをPCに転送を行ないたい場合、PCにおいて転送用のアプリケーションを用いて、US0001.MPGを取得する命令を発行する。

【0047】図13にシステム構成図の一例を示す。この図では、ビデオカメラとPCがIEEE1394で接続されている。それぞれのシステムにおいて、ディスク媒体を管理している論理ファイルシステムを元にデバイスドライバが用意されている。つまり、それぞれのユーザシステムがそれぞれのデバイスドライバを介して、ファイル単位のディスクへのアクセスができることになる。また、それぞれのシステムにはIEEE1394を使用するためのドライバが用意されている。

【0048】例えば、PC側のユーザシステムB(7)がIEEE1394ドライバB(8)を介して、ビデオカメラ側のIEEE1394ドライバA(3)に対してユーザシーンを管理する仮想ファイル名を指定してストリームデータの取得要求を出すと、IEEE1394ドライバA(3)はデバイスドライバA(2)に対して指定された仮想ファイルの読み出し命令を出す。これにより、ディスク(6)上のストリームデータがディスクから読み出され、IEEE1394ドライバA(3)を経由してPC側に転送される。転送されたデータはPC側のIEEE1394ドライバB(8)を経由しユーザシステムB(7)の指定するファイル名でデバイスドライバB(9)を介してPC側のディスク(12)上に記録される事になる。

【0049】転送したストリームデータを再生する場合は、PCにおいて再生するためのアプリケーションプログラムを用いて、転送したストリームデータを読み込む。従来、ディスクから読み出したストリームデータを端から順番に最後まで再生を行なうと、前述したようにEUあるいはVU単位での再生となる。しかし、実際にはユーザによる編集によって定義されたユーザシーンはEUやVUの途中のフレームから再生を開始し、最後のEUやVUに関しても途中のフレームで再生を終了するものである可能性が高い。

【0050】図14において、ユーザが指定したユーザシーンの再生範囲は、EU#0中のVU#3中の6番目の映像フレーム（フレーム番号5）からEU#20内のVU#121の中の12番目のフレーム（フレーム番号11）までの例を示している。つまり、転送先のディスクでのストリームデータはEU#0～EU#20までであり、このストリームデータ全体を再生すると、ユーザによって指定されたユーザシーンの再生範囲と異なっている。

【0051】そこで、本実施形態においては、ユーザシーンを正確に再生するための情報をストリーム上のUnit Headerに記録することで、ストリームデータのみで、

フレーム単位での再生範囲の指定を行うことを可能とする。

【0052】Unit Headerパケットにユーザシーンを正確に再生するための情報として、Number of ScenesとScene Informationがある。Scene InformationはNumber of Scenesの数だけ記録されることになる。Scene Informationの詳細を以下に示す。

【0053】Scene Nameは、シーンの名前を格納し、Scene Creation Dateはシーンの作成日時、Scene Data Sizeは、定義されたシーンを管理するファイルのデータサイズを格納する。このデータサイズはディスクから読み出した後の実データサイズを表すものである。

【0054】Scene Start VU Numberは、ファイルとして管理されているストリームデータの中で再生を開始したい映像フレームが含まれるVUの番号をストリームデータの先頭のVUを基準とした値で管理する。PRUが存在しないストリームの場合は、必ずこの値は0となり、PRUが存在するストリームの場合は、EUの中の再生を開始したいフレームの含まれるVUの番号を管理することになる。

【0055】Scene Number of VUは、ユーザシーン中のVUの数を示すための情報であり、Scene Start VU Numberで管理されるVUから再生を停止するフレームの含まれるVUまでのVU数を管理する。

【0056】Scene Start Frame Numberは、再生を開始するフレームの番号を管理する。フレーム番号はScene Start VU Numberで管理される再生開始VU番号内のフレーム番号となる。つまり、再生を開始する映像フレームは、ディスクから読み出したストリームデータ中のScene Start VU Number目のVU内のScene Start Frame Numberからとなる。

【0057】Scene End Frame Numberは、再生を停止するフレームの番号を管理する。フレーム番号はScene Start VU Numberで管理される再生開始VU番号にユーザシーン中のVU数であるScene Number of VUを足した値から1を引いたものに相当する再生終了VU内のフレーム番号になる。

【0058】以上説明したようなUnit Headerを使うことによって、ユーザシーンの正確な再生が可能となり、ユーザシーンに関する管理情報を含める事が可能となる。実際には、PRUが存在するストリームであっても存在しないストリームであっても、必ずストリームデータの先頭はUnit Headerから始まることになる。よって、これから再生しようとするストリームデータの一番先頭のUnit Headerを読み出すことによってユーザシーンの再生情報を把握することが可能となる。つまり仮想ファイルによって管理されるストリームデータにおいて、一番先頭にVUがくる場合（PRUが定義されていない場合）はVUに対応するUnit HeaderにScene Informationを記録し、PRUが先頭に配置される場合はPRUに対応するUnit HeaderにScene Informationを記録することになる。

【0059】図14の例では、PRUが定義されているストリームデータが仮想ファイルUS0001.MPGによって管理されており、一番先頭にPRUが配置されているのでPRUの先頭に存在するUnit Header中のScene Informationにシーンの再生情報を格納することになる。この例では、再生を開始するフレームが含まれるVU番号を管理するScene Start VU Numberには3を、シーン中のVU数を管理するScene Number of VUには118を、再生開始フレーム番号を管理するScene Start Frame Numには5を、再生を停止するフレーム番号には11が格納される事になる。また、図15にPRUが存在しないストリームデータが仮想ファイルUS0001.MPGによって管理されている様子を示す。この例ではPRUが定義されていない関係で、1EUが1VUによって構成されており、仮想ファイルで管理されるストリームデータの先頭はVU#0の先頭に存在するUnit Headerとなり、このUnit Headerにこのユーザシーンの管理情報を格納する事になる。

【0060】このように構成した場合、複数のユーザシーンが同一のEUから再生を開始する場合がある。このような場合、図16に示すように、例えばユーザシーンUS0001.MPGとUS0002.MPGが同一のEUから再生を開始する場合に、Unit Headerに格納されたどのScene Informationを使用したら良いのかが不明になる。そこで、Scene Information中のScene名をユーザに提示して再生情報を選択すると言った事が可能である。例えば、Scene名として仮想ファイルのファイル名を使うことも考えられるが、この場合、仮想ファイルのファイル名が変更になると対応関係の整合性が取れなくなってしまう。PRUが定義されないストリームの場合は、ユーザシーンはEU単位ではなくVU単位で構成され、図16におけるEUを全てVUに置き換えて考えれば良いので説明を省く。

【0061】自動的にSceneに対応する再生情報を選択させたい場合は、転送を行なったファイルシステムで管理されるファイルのデータサイズとScene Information内のScene Data Sizeの値を比較することによって、目的の再生情報を選択することが可能となる。図16の例では、ユーザシーンUS0001.MPGはEU#0～EU#nで構成されており、ユーザシーンを管理する仮想ファイルのファイルサイズが256000KBである。また、ユーザシーンUS0002.MPGはEU#0～EU#n-3で構成されており、ユーザシーンを管理するファイルサイズは20480KBである。Unit Headerに記録されるScene InformationのScene Data Sizeにそれぞれ、256000KBと20480KBが記録されたフィールドがあるかどうかを調べることによって、対応するScene Informationを自動的に抽出することができる。

【0062】ここで、具体的な処理として、ビデオカメラで撮影を行ないオリジナルシーンをファイルとしてディスクに記録した場合について説明する。オリジナルシーンをディスク上に記録し、前述の論理ファイルシステムのFile Descriptorが作成されるまでの手順は従来と

同様であるので、省略する。ここでは、ファイルが記録された後の本発明特有の処理について、図17のフローチャートを用いて説明する。

【0063】ステップS10においてオリジナルシーンの作成要求があり、ストリームデータを管理するファイルの作成が終了したら、ステップS11において、ファイルによって管理されているストリームデータのディスク上での位置情報を元に、対応するストリームデータの先頭の論理ブロックを読み出す。論理ブロックとは、ファイルシステムで管理する最小読み書きの単位である。こ

こで、読み出した論理ブロックは、オリジナルシーンの先頭に配置されているUnit Headerが記録されているUnit Headerバケットとなる。このUnit Headerにオリジナルシーンの再生情報を記録することになる。

【0064】ステップS12において、Unit Header内のNumber of Sceneに1をセットする。ステップS13において、Scene Information内のScene Nameにシーン名をセットし、Scene Creation Time and Dateオリジナルシーンの作成日時をセットする。ステップS14において、Scene Information内のScene Data Sizeにファイルシステムで管理しているオリジナルシーンのファイルサイズ、つまりFile Descriptorに記録されているファイルのデータサイズをセットする。

【0065】ステップS15において、Scene Information内のScene Start VU Numberに0を、Scene Number of VUにオリジナルシーン中のVU数の値を、Scene Start Frame Numberに0を、Scene End Frame Numberにオリジナルシーン中の最後のVUの最後のフレーム番号をセットする。フレーム番号はVU中の番号であり0から始まる番号である。ステップS16において、ステップS12からS15において更新した内容をディスク上に更新し処理を終了する。ここまでで、オリジナルシーンの先頭のUnit Headerにオリジナルシーンとしての再生情報が作成されたことになる。

【0066】次に、ユーザがオリジナルシーンを素材としてユーザシーンを定義した場合を説明する。ユーザシーンに対応する映像データは仮想ファイルの先頭のユニット中の任意のフレームから、仮想ファイルの最終のユニットの任意のフレームまでである。そこで、ユーザが任意のシーンを指定した場合、その先頭のフレームを含むユニットからその最後のフレームを含むユニットまでを仮想ファイルとして、まず定義する。この仮想ファイルの定義については、上記したものと同様であり、Virtual File Descriptorが作成される。Virtual File Descriptorが作成された後の処理について、以下に図18のフローチャートを元に説明する。

【0067】ステップS20において、ユーザシーンの作成要求があり、対応するストリームデータを管理する仮想ファイルの作成（File Descriptorの作成）が終了したら、ステップS21において、仮想ファイルによって管

理されているストリームデータのディスク上での位置情報を元に、対応するストリームデータの先頭の論理ブロックを読み出す。読み出した論理ブロックは、ユーザシーンの先頭に配置されているUnit Headerが記録されているUnit Headerバケットである。このUnit Headerにユーザシーンの再生情報が記録されることになる。

【0068】ステップS22において、Unit Header内のNumber of Sceneの値に1を足す。ステップS23において、対応するScene Information内のScene Nameにシーン名をセットし、Scene Creation Time and Dateユーザシーンの作成日時をセットする。ステップS24において、Scene Information内のScene Data Sizeにファイルシステムで管理しているユーザシーンのファイルサイズ（Virtual File Descriptor中のファイルサイズ）をセットする。

【0069】ステップS25において、Scene Information内のScene Start VU Numberにユーザシーンのストリームデータ中の再生を開始するフレームが含まれるVU番号をセットする。VU番号は、ユーザシーンのストリームデータの先頭を基準として相対的な値である。ステップS26において、Scene Information内のScene Number of VUにユーザシーン中の再生するVU数の値をセットする。ステップS27において、Scene Information内のScene Start Frame Numberに、再生を開始するフレームの番号をセットする。フレーム番号は、ステップS25においてセットしたScene Start VU Numberで管理されるVU内のフレーム番号であり、VUの先頭を基準として相対的な値である。ステップS28において、Scene Information内のScene End Frame Numberにユーザシーン中の表示する最後のVU内のフレーム番号をセットする。表示する最後のフレームが含まれるVU番号は、ステップS25においてセットした、Scene Start VU NumberにステップS26においてセットしたScene Number of VUの値-1を足し合わせた値となる。ステップS29において、ステップS22からS28において更新したUnit Headerの内容をディスク上に更新し処理を終了する。

【0070】図14、15は上記した図18のフローチャートに基づいて、ユーザシーンに関する再生情報が記録された様子を示している。

【0071】仮想ファイルで管理するユーザシーンに対応するストリームデータをネットワークなどを介してPC等に転送した際に、転送先のPCでのユーザシーンの正確な再生方法について図19に示すフローチャートを元に説明する。もちろん転送しない場合の再生についても同様である。

【0072】ステップS30において、ユーザシーンの再生要求が発生すると、ステップS31において、ユーザシーンを管理するファイルのディスク上の位置情報からストリームデータの先頭の論理ブロックを読み出す。読み出した論理ブロックは、Unit Headerに相当するUnit He

10

20

30

40

50

aderバケットである。読み出したUnit HeaderからNumber of Scenesの値を把握する。ステップS32において、Unit Header内のNumber of Scenesが0の場合は、ステップS33において、表示開始フレームをストリームデータ中の最初の表示フレームに、表示終了フレームをストリームデータ中の最後の表示フレームにセットする。つまり、ストリームデータ中の全ての映像フレームを表示することになる。Number of Scenesが値が0の場合というのはユーザシーンを作成した際に、何らかの理由によって正しくScene Informationが記録されなかったイリーガルな状況である。ステップS35でストリームデータからデータを読み出して、デコードを行い再生する。ここでは全てのフレームが再生されることになる。

【0073】ステップS32において、Unit Header内のNumber of Scenesが1の場合は、ステップS34において表示開始フレームを、Scene Information内のScene Start VU Numberで示されるVU内のScene Start Frame Numberで示されるフレームとし、表示終了フレームを、Scene Start Frame NumberとScene Number of VUを足し合わせた値-1のVU内のScene End Frame Numberで示されるフレームに設定する。

【0074】ステップS35において、ステップS34において設定された再生情報を元にディスクから読み出したストリームデータに対して表示の制御を行ない処理を終了する。具体的には、ディスクから読み出されるストリームデータは、EU単位で読み出され、VU単位にデコードされ、再生制御情報に基づいて開始フレームから再生が行われる。

【0075】ステップS32において、Unit Header内のNumber of Scenesが2以上の場合は、ステップS36においてUnit Header内のすべてのScene Informationをチェックしたかどうかを判断する。ステップS36において全てのScene Informationのチェックが終わっていなければ、ステップS37においてファイルシステムで管理されているユーザシーンのファイルサイズとステップS31において読み出したScene Information内のScene Data Sizeを比較する。ステップS38においてユーザシーンを管理しているファイルのファイルサイズと、Scene Information内のScene Data Sizeが一致する場合は、ステップS34において表示開始フレームを、Scene Information内のScene Start VU Numberで示されるVU内のScene Start Frame Numberで示されるフレームとし、終了フレームをScene Information内Scene Start VU NumberにScene Number of VUを足し合わせた値-1のVU内のScene End Frame Numberと設定する。ステップS35において、ステップS34において設定された再生情報を元にディスクから読み出したストリームデータに対して表示の制御を行ない処理を終了する。

【0076】ステップS38においてユーザシーンを管理している仮想ファイルのファイルサイズと、Scene Info

rmation内のScene Data Sizeが一致しない場合は、ステップS36に戻り次のScene Informationについて同様の処理を繰り返す。ステップS36において全てのScene Informationのチェックが終わった場合は、該当するシーンがないため、イリーガルな状態ではあるが、一例としてS39においてユーザに対してScene Informationを選択させる事も可能である。あるいは、S39の段階でエラー処理をして処理を終了しても良い。

【0077】続いて、第2の実施形態として第1の実施形態で説明してきた再生情報を、ユーザ定義したヘッダ用のバケットを使用せずMPEG規格準拠のヘッダ領域に格納する場合について説明する。

【0078】MPEG1およびMPEG2の符号化されたストリームデータは、図20のようにシーケンス層、GOP層、ピクチャ層、スライス層、マクロブロック層、ブロック層と言った階層構造になっている。それぞれの階層においてヘッダ情報が付加されており、ここではシーケンス層およびGOP層に注目する。

【0079】ここで、MPEG2規格におけるビデオストリームのシーケンスの定義を図21に示す。この図のようにシーケンス層において、シーケンスヘッダの情報(sequence-header() 20)、シーケンスエクステンション(sequence-extension() 21)の後に、ユーザデータ(extension-user-data(0) 22)を記録することが可能である。図22にextension-user-data()の処理内容を示す。ここで、user-data-start-code 30が記録されているかを判定する。user-data-start-codeは16進数で000001B2である。図23に示すユーザデータの内容は、8bit*n個単位の情報を格納することができ16進数で000001、つまり次の情報のstart code 40が現れるまで記録できる。またGOP層において、GOPヘッダの情報(group-of-pictures-header() 23)の後にシーケンス層と同様にユーザデータ(extension-user-data(1) 24)を記録することが可能である。ユーザデータの形態はシーケンス層の場合と同様である。

【0080】このようにシーケンス層およびGOP層にユーザ定義の情報を記録することが可能となっている。ここで、第1実施形態において説明してきた再生情報をUnit Headerではなく、このユーザデータ領域に格納することを考える。

【0081】ここで、この再生情報がGOP単位で必要になることを考えた場合に、ストリーム構成に関して若干の制限を付ける必要がある。まず、MPEG2規格においてGOP層のヘッダ情報を持たないストリームを定義することが可能であるが、本実施形態においては、かならず1つのGOPに対して1つのGOP層のヘッダ情報を付加するようにする制限を付ける。

【0082】本実施形態においては、GOP層におけるユーザデータに再生情報を格納する場合について説明する。前述のGOP層におけるextension-user-data(1)で再

生情報を格納する。再生情報自体は第1実施形態で述べた構成と同様なので説明は省略する。具体的にはUnit Header中のNumber of ScenesとScene Informationの値をGOP層のユーザデータに記録することになる。

【0083】GOP層のユーザデータにおいて記録できるScene Informationの数の上限値を定める。例えば上限を20個と定め、最初にストリームを記録する際に、GOP層におけるユーザデータを記録する領域としてScene Informationを20個記録できる領域をあらかじめ確保しておく。最初にストリームを記録する時に、ダミーデータ例えば16進数でFFをこの領域に記録することによって、後からユーザの編集により再生情報を追加する際に記録するデータ領域が無いといった事を防ぐ事が可能となる。

【0084】このように定義されたビデオのストリームデータとオーディオストリームデータを多重し、GOP構造に相当する管理単位毎にランダムアクセスを行なうものとする。ユーザシーンを管理する仮想ファイルは、ユーザシーンの再生開始および終了フレームを含むGOP構造に相当する管理単位毎の位置情報を管理すれば良い。

【0085】ここで、このストリームデータをディスクに記録する場合、上記第1の実施形態のように、各GOP層のヘッダ情報がディスクの論理ブロックにアライメントされるように制御されている場合であれば、ユーザシーンを管理している仮想ファイルの先頭の論理ブロックを読み出せば、GOPヘッダを読み出し、更新することが容易にできる。しかし、ストリーム中の各GOPのヘッダ情報がディスクの論理ブロックにアライメントされていない場合は、ユーザシーンを管理している仮想ファイルの先頭の論理ブロックを読み出しても、必ずしも注目しているGOPヘッダの情報が読み出した論理ブロックの先頭に入っているとは限らない。つまり、注目している1つ前のGOPの情報が読み出したデータの先頭に入っていることも考えられる。

【0086】更に、実際にMPEGストリームデータをディスクに記録する際はMPEGのシステム規格であるPESストリームやPSストリームなどと言ったバケット形式に整形してから行なうのが一般的である。PESやPSのバケットの形式で記録されるという事は、基本的にビデオストリームとオーディオストリームを多重したストリームデータを任意の大きさに分割しバケット化し、それぞれのバケットに対してヘッダ情報を付けるものである。

【0087】このような場合は、読み出したストリームデータを先頭から順番に見ていき、余分なヘッダ情報を読み飛ばして、GOP層のヘッダ情報であるextension-user-data(1)が見つかるまでサーチすることになる。具体的には、user-data-start-codeとして16進数で000001B2が見つかるまでサーチをする。サーチ操作で特定した再生情報を記録する領域を読み出したり、更新したりすることになる。

【0088】再生情報の書き込みなどの処理手順に関し

ては、第1実施形態で説明した場合と基本部分では同一なので説明を省略する。

【0089】第1および第2実施形態において、ユーザシーンを仮想ファイルで管理する前提で説明してきたが、図26に示した従来方法のようにユーザシーンが参照するオリジナルシーンの任意箇所に対応するディスク上の位置情報を1つの管理情報ファイルとして管理し、再生情報であるScene Informationをその参照しているストリームデータの先頭のUnit Headerに入れるようにしても良い。このような方法によって仮想ファイル機能が無いようなシステムにおいても、本発明を実施する事が可能となる。この場合、ユーザシーンを他の機器などにコピーや転送したい場合、管理情報ファイルで管理されているユーザシーンのディスク上の位置情報を元に、ディスクからストリームデータを読み出す位置を特定し転送を行う。転送先では、再生情報がストリームデータの先頭のUnit Headerに格納されているストリームデータがファイルとして管理される事になる。

【0090】

【発明の効果】本発明によれば、映像データ及び音声データが記録されているユニットの補助データに再生開始位置及び再生終了位置を示す情報を格納可能としているために、オリジナルシーンの一部をユーザシーンとする場合に、そのユーザシーンの先頭の補助データに再生情報を格納することで、ユニット中の任意の位置からの再生開始、再生終了制御を行うことが可能であり、かつ、再生情報を例えば他のファイルなどで管理する必要がない。

【図面の簡単な説明】

【図1】本発明のデータ管理方法の実施形態が対象とする装置の構成を示すブロック図である。

【図2】本発明のデータ管理方法の実施形態で扱うPRUが定義されたMPEGストリームの構成を示す図である。

【図3】本発明のファイル管理方法の実施形態で扱うPRUの定義されていないMPEGストリームの構成を示す図である。

【図4】本発明のデータ管理方法の実施形態で扱うPRUの定義されたMPEGストリームとブロックの関係を示す図である。

【図5】本発明のデータ管理方法の実施形態で扱うPRUの定義されていないMPEGストリームとブロックの関係を示す図である。

【図6】本発明のデータ管理方法の実施形態で扱うUnit Headerの内容を示す図である。

【図7】本発明のデータ管理方法の実施形態において1つの仮想ファイルがオリジナルシーンのデータを参照している様子を示す図である（PRUありの場合）。

【図8】本発明のデータ管理方法の実施形態において2つの仮想ファイルがオリジナルシーンのデータを参照している様子を示す図である（PRUありの場合）。

【図 9】本発明のデータ管理方法の実施形態におけるディスクに記録された実ファイルの論理ファイルシステムの管理情報と実ファイルで管理されるデータの関係を示す図である。

【図 10】本発明のデータ管理方法の実施形態におけるディスクに記録された仮想ファイルの論理ファイルシステムの管理情報と仮想ファイルで参照されるデータの関係を示す図である。

【図 11】本発明のデータ管理方法の実施形態において 1つの仮想ファイルがオリジナルシーンのデータを参照している様子を示す図である（PRUなしの場合）。

【図 12】本発明のデータ管理方法の実施形態において 2つの仮想ファイルがオリジナルシーンのデータを参照している様子を示す図である（PRUなしの場合）。

【図 13】本発明のファイル管理方法の実施形態におけるシステム構成を示す図である。

【図 14】本発明のデータ管理方法の実施形態において仮想ファイルが管理するストリームデータとユーザが実際に指定した再生範囲の関係を示す図である（PRUありの場合）。

【図 15】本発明のデータ管理方法の実施形態において仮想ファイルが管理するストリームデータとユーザが実際に指定した再生範囲の関係を示す図である（PRUなしの場合）。

【図 16】本発明のデータ管理方法の実施形態において同一のEUから再生を開始する仮想ファイルの様子を示す図である。

【図 17】本発明のデータ管理方法の実施形態においてオリジナルシーンを作成する際に行なう処理の流れを示すフローチャートである。

【図 18】本発明のデータ管理方法の実施形態においてユーザシーンを作成する際に行なう処理の流れを示すフローチャートである。

*

【図 19】本発明のデータ管理方法の実施形態においてユーザシーンを再生する際に行なう処理の流れを示すフローチャートである。

【図 20】本発明のデータ管理方法の第2の実施形態において扱うMPEGの階層構造を示す図である。

【図 21】本発明のデータ管理方法の第2の実施形態において扱うMPEGのビデオシーケンスの処理を示す図である。

【図 22】本発明のデータ管理方法の第2の実施形態において扱うMPEGの拡張およびユーザデータの処理を示す図である。

【図 23】本発明のデータ管理方法の第2の実施形態において扱うMPEGのユーザデータの処理を示す図である。

【図 24】従来技術における実ファイルと仮想ファイルの関係を示す図である。

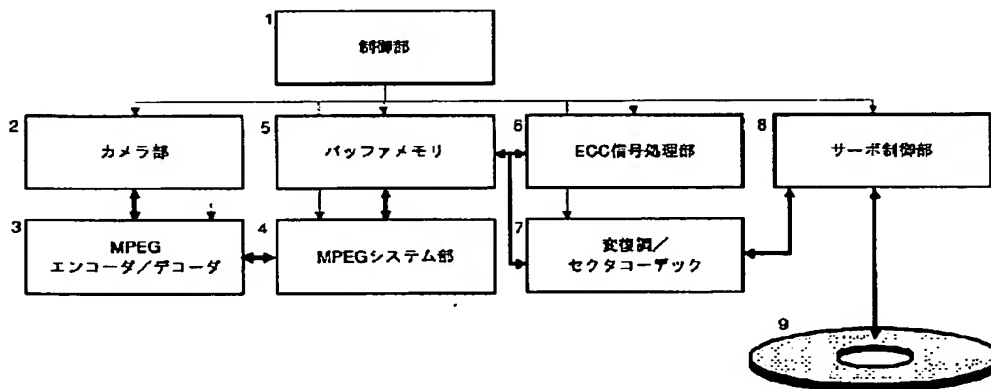
【図 25】従来技術におけるユーザシーン毎に再生情報を持ちそれぞれ実ファイルとして管理されている様子を示す図である。

【図 26】従来技術におけるストリームデータを実ファイル出管理しそのストリームデータの任意の箇所を参照するユーザシーンの再生情報を管理情報として実ファイルで記録する様子を示す図である。

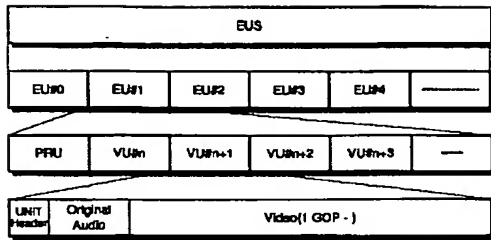
【符号の説明】

- 1 制御部
- 2 カメラ部
- 3 MPEGエンコーダ
- 4 MPEGシステム部
- 5 バッファメモリ
- 6 ECC信号処理部
- 7 変復調/セクタコーデック
- 8 サーボ制御部
- 9 ディスク

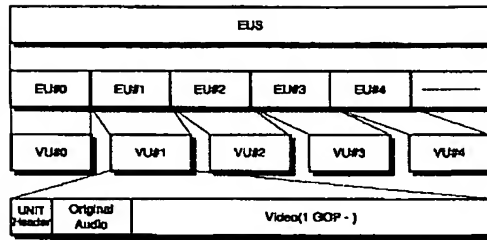
【図 1】



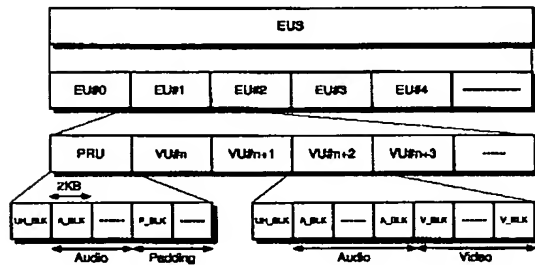
【図2】



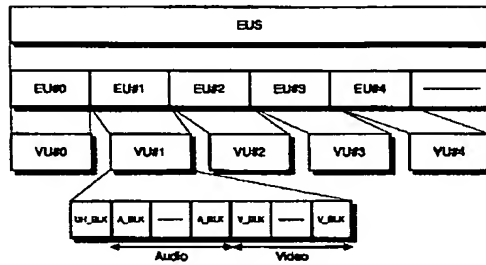
【図3】



【図4】



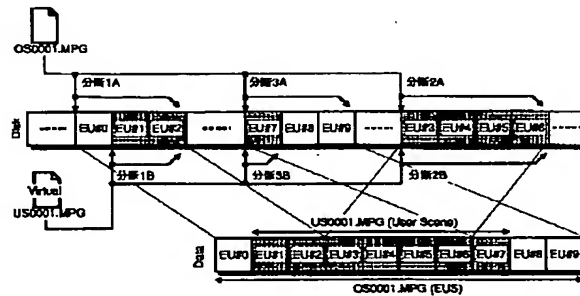
【図5】



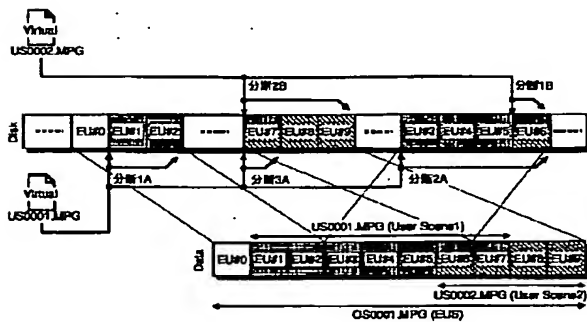
【図6】

BP	Length	Field Name	Contents
0	3	packet_start_code_prefix	000001h
1	1	stream_id	10111111b
2	2	PES_packet_length	Unit6
4	1	Unit_Property	Unit6
5	2	Length of Unit	Unit6
7	2	Start FLBN of Video Data	Unit6
9	1	Number of IP Pictures	Unit6
11	1	End FLBN of IP Pictures	Unit6
12	2	Start FLBN of VU	Unit6
14	1	Number of VU	Unit6
15	1	Start FLBN of Data for VU	Unit6
16	1	Number of Scenes	Unit6
17	2	Scene Information	Unit6
19	(12)	Scene Name	string
21	(4)	Scene Creation Time and Date	Unit32
23	(4)	Scene Data Size	Unit32
25	(1)	Scene Start VU Number	Unit6
26	(4)	Scene Number of VU	Unit32
28	(1)	Scene Start Frame Number	Unit6
29	(1)	Scene End Frame Number	Unit6

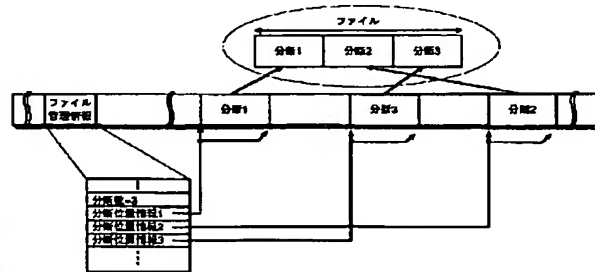
【図7】



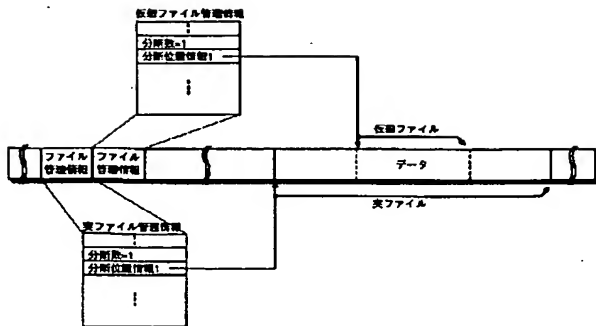
【図8】



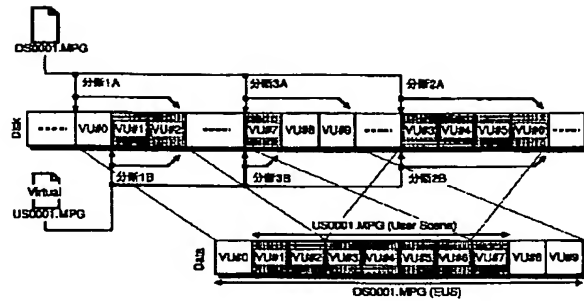
【図9】



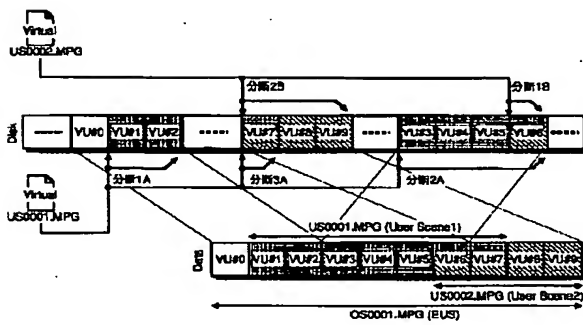
【図10】



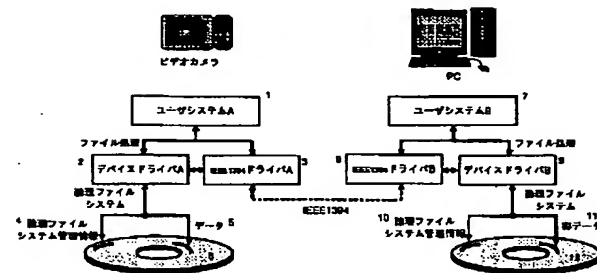
【図11】



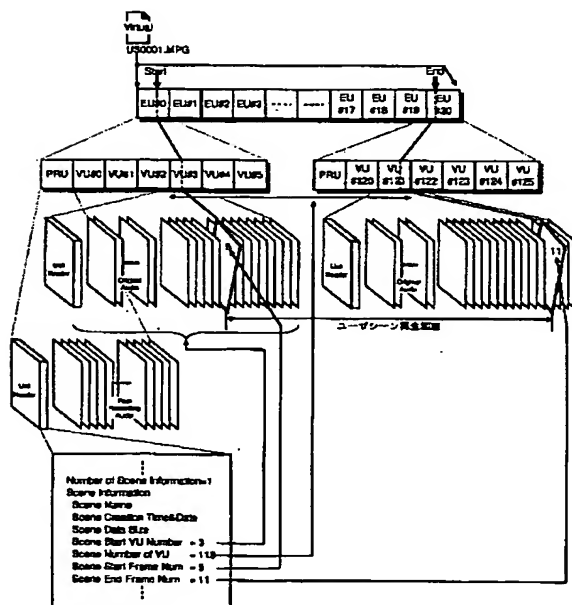
【図12】



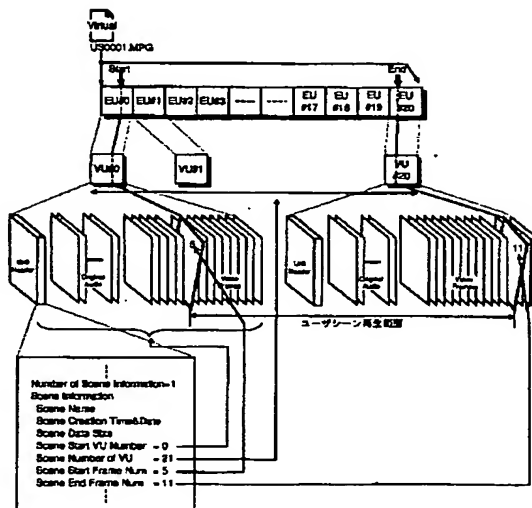
【図13】



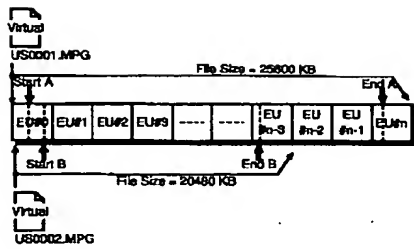
【図14】



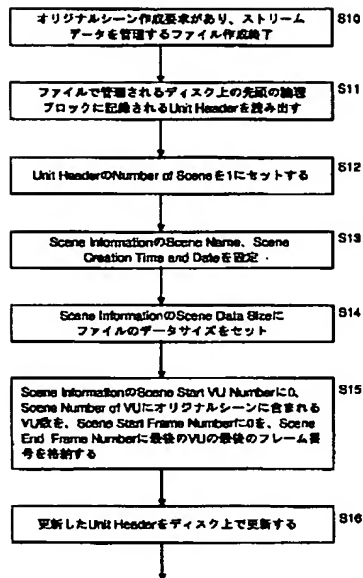
【図15】



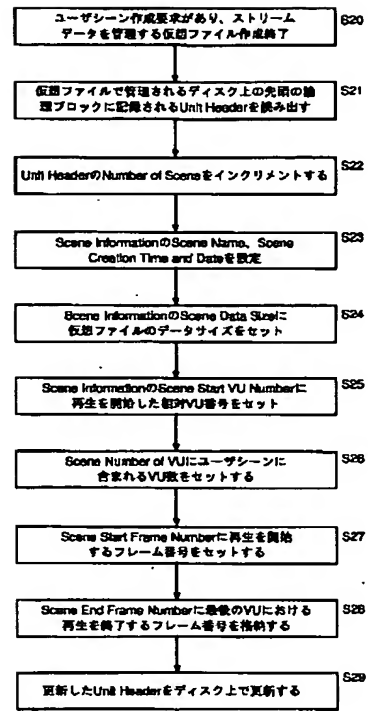
【図16】



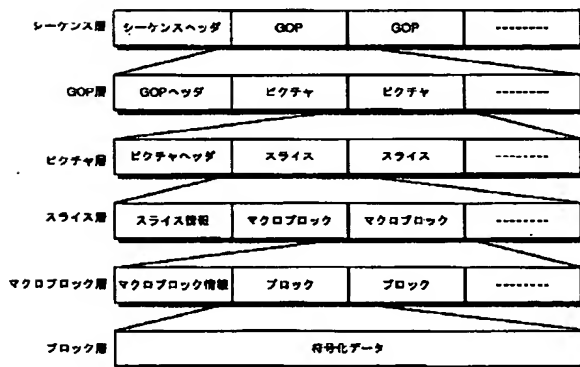
【図17】



【図18】



【図20】



【図21】

```

video_sequence()
{
    next_start_code()
    sequence_header()
    if (nextbits() == extension_start_code)
    {
        sequence_extension()
        do
        {
            extension_and_user_data()
        }
        while (nextbits() == group_start_code)
        {
            group_of_pictures_header()
            extension_and_user_data(1)
        }
        picture_header()
        picture_coding_extension()
        extensions_and_user_data(2)
        picture_data()
        while (nextbits() == picture_start_code)
        {
            nextbits() == group_start_code
        }
        while (nextbits() == sequence_end_code)
        {
            sequence_header()
            sequence_extension()
        }
        while (nextbits() == sequence_end_code)
    }
    else
    {
        ISO11172-2
    }
    sequence_end_code
}

```

【図22】

```

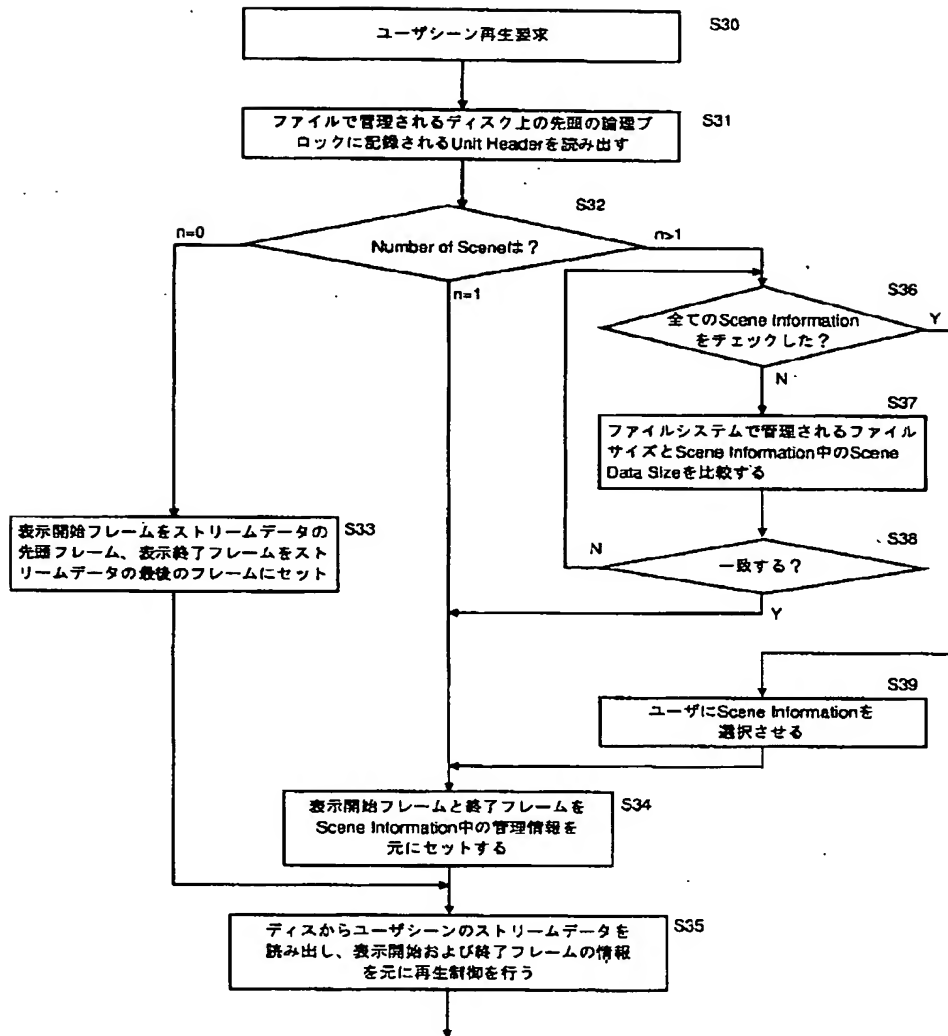
extension_and_user_data()
{
    while ((nextbits() == extension_start_code) ||
           (nextbits() == user_data_start_code))
    {
        if (nextbits() == extension_start_code)
        {
            extension_data()
        }
        if (nextbits() == user_data_start_code)
        {
            user_data()
        }
    }
}

```

【図25】



【図19】



【図23】

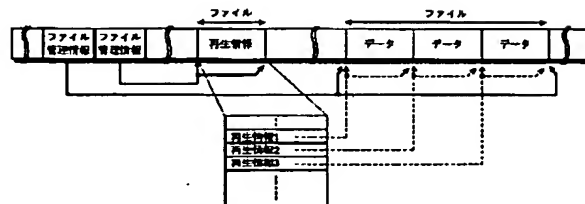
```

user_data{
  user_data_start_code
  while (nextbits() != "0000 0000 0000 0000 0001") {
    user_data
  }
  next_start_code()
}

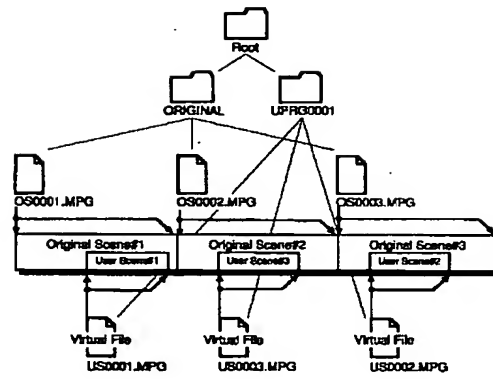
```

40

【図26】



【図24】



フロントページの続き

(51)Int.Cl.⁷

H04N 5/91

識別記号

F I

G 1 1 B 27/02

テーマコード (参考)

A